

## 4G network에서 재구성성을 위한 개방형 API

홍성준\*, 이용수\*\*

### The open API for reconfiguration in 4G network

Sung-June Hong\*, Young-Soo Lee\*\*

#### 요약

본 논문은 4G 이동 네트워크 환경에서 재구성성을 제공하기 위한 방안으로 PARLAY X 기반의 개방형 API에 관하여 서술하였다. 4G 이동 네트워크 환경에서는 망 수준에서 문맥인식성, 적응성, 개인화 등과 같은 지능성이 요구된다. 그러나 기존의 PARLAY 기반 서비스 네트워크는 문맥인식성, 적응성, 개인화 등의 지능성에 대한 고려가 부족하다. 그러므로 본 논문의 목적은 4G 이동 네트워크 환경에서 문맥인식성, 적응성, 개인화 등이 지원 가능한 개방형 API를 개발하고자 하는 것이다. 본 논문에서 개방형 API는 사용자 주변에 변화하는 문맥 제약조건 및 사용자 각 개인의 선호도 정보를 파악하여, 변화하는 문맥 제약 조건 및 사용자의 선호도에 따라서 각 개인에게 적합한 적응성 있는 망 서비스를 제공하는 것을 의미하며, 개방형 API를 이용하는 예로 위치와 속도 등의 문맥정보와 사용자의 선호도 정보를 파악한 후에, 사용자에게 망에서 문맥정보 및 사용자 선호도에 따라서 각 개인에게 적합한 차별화된 QoS를 지원하는 경우를 보였다.

#### Abstract

This paper describes the open API with integration of semantic web service with PARLAY X based open API in 4G mobile network. It can be expected that the intelligence such as the context-awareness, adaptation and personalization in 4G mobile network will be deployed. But the existing PARLAY based network lacks in considering context-awareness, adaptation and personalization. Therefore, the object of this paper is to support the architecture and the Application Programming Interface (API) of the network service for the context-awareness, adaptation and personalization in 4G mobile network. The open API is to provide users with the adaptive network service to the changing context constraints as well as detecting the changing context and user's preference. For instance, the open API can provide users with QoS in network according to the detected context and user's preference, after detecting the context such as location and speed and user's preference.

▶ Keyword : Open API, Reconfiguration, 4G, QoS(Quality of Service)

• 제1저자 : 홍성준

• 접수일 : 2005.10.10, 심사완료일 : 2005.12.15

\* 여주대학 정보통신과 조교수, \*\* 여주대학 컴퓨터정보관리과 부교수

※ 본 연구는 2005년도 여주대학 교내 연구비 지원으로 수행된 과제임.

## 1. 서론

최근 3G 이동 네트워크 환경의 도래와 함께 4G 이동 네트워크에 대한 관심이 증가되고 있으며, 4G 이동 네트워크 환경은 빠르게 변화되는 환경에 적응할 수 있는 유연한(seamless) 응용 프로그램의 요구가 대두되고 있다.

이러한 요구에 부응하여 망 측면에서는 PARLAY(1)라는 개방된 네트워크 서비스 전송 플랫폼 표준이 제정되었다. PARLAY는 1998에 망 오퍼레이터, 망 벤더로 구성되었고 일반 프로그래머가 네트워크를 사용하거나 부가 가치 서비스를 더 지원할 수 있다는 생각으로 시작되었다. PARLAY는 전화통신망과 IT기술, 인터넷 그리고 프로그래밍 패러다임을 통합하기 위한 개방된 API(Application Programming Interface)이다. PARLAY 표준화 단체에서는 일반 개발자의 응용 프로그램이 안전하고 실시간으로 통신망 자원을 사용할 수 있도록 해주는 API(Application Programming Interface) 제정 활동을 하고 있다. 특히 PARLAY X는 XML기반의 웹 서비스 기술을 이용하여 네트워크 프로토콜 추상화로 서비스 프로그래밍을 지원한다.

한편, 기술 상황이 지속적으로 변화하기 때문에 특정 기술에 기반된 시스템을 기술 변화에 맞게 통합, 변화, 유지하는 것은 여전히 문제로 남아있다. 이러한 문제점을 해결하기 위해서 OMG(Object Management Group)는 MDA(Model Driven Architecture){2}{3}라는 개발 패러다임을 제시하였다. MDA는 소프트웨어, 하드웨어, 사람, 비즈니스를 구성하는 기업 환경 시스템의 통합과 상호 운영성에 대한 접근 방법이며, 기업 시스템의 모든 측면을 이해하고 설계하고 동작하도록 하는 프레임 워크를 제공한다. 결과적으로 PARLAY 서비스 전송 플랫폼에서 빠르고 자동화된 네트워크 서비스를 위한 MDA기반의 SCE(Service Creation Environment){4}를 사용하는 것이 필수적이며, MDA기반의 SCE를 지원하는 도구로는 medini{5} 등이 있다.

4G 이동 네트워크 환경에 적합한 네트워크 서비스를 위해서는 문맥인식성{6}{7}{8}, 적응성, 개인화 등과 같은 망에서의 지능성 지원이 요구되고 있다. 그러나 기존의 PARLAY X API는 이러한 문제를 고려하지 못하였다. 이러한 문제를 해결하기 위해서 3G네트워크상에서 시멘틱 웹 서비스 기술을 이용하는 WASP라 불리는 프로젝트가 제안

되었다. 그러나 WASP는 너무 시멘틱 웹 서비스 기술{9}{10}에 의존하여 시멘틱 웹 서비스 기술에 익숙하지 않은 개발자들이 새로운 망 서비스를 개발하는데 많은 어려움이 있다. 즉 기존의 PARLAY의 문제점으로 4G 무선 네트워크 환경에서 문맥인식성, 적응성, 개인화 지원에 대한 고려가 부족하다. 그러므로 본 논문의 목적은 문맥인식성, 적응성, 개인화의 지원이 가능한 PARLAY X기반의 신규 망 서비스 구조 및 PARLAY X API를 개발하고자 하는 것이다. 본 논문의 장점은 첫째, 기존의 PARLAY X에서의 지원 범위를 넘어서는 문맥인식성, 적응성, 개인화를 표현하고 지원할 수 있다. 둘째, 문맥인식성과 적응성, 개인화와 같은 지능성을 망에서 제공하므로 제 4세대 무선 통신망이나 BcN(Broadband convergence Network)기반의 차세대 통신망에서의 기술적 요구사항을 지원할 수 있다.

본 논문에서 제안한 개방형 API는 사용자에게 변화하는 문맥 제약 조건 및 사용자의 선호도에 따라서 각 개인에게 적합한 적응성 있는 망 서비스를 제공하는 것을 의미한다. 한 예로, 개방형 API를 이용한 시나리오에서는 무선 게임을 즐기는 상황에서 위치를 이동하는 경우를 가정하였고, GPS나 센서가 사용자의 위치나 속도 정보와 사용자의 선호도를 인식한 후, 그 위치와 속도 정보 및 사용자 선호도 정보를 개방형 API에게 알려주면 개방형 API는 망 자원 상태를 파악하고 그 지역의 망 자원 상태에 적합한 차별화된 등급의 서비스를 전송하도록 하여 망에서 각 개인에게 적합한 QoS(Quality of Service)를 사용자에게 제공한다. 개방형 API의 구조는 SCE와 Semantic Context Broker, 그리고 PARLAY G/W로 구성되어 있다. SCE는 기존 지능망의 망 서비스와 같은 망 서비스로서 문맥인식성과 적응성을 표현하고 지원하기 위한 망 서비스 개발 환경을 의미한다. 본 SCE는 OCL(Object Constraint Language){11}을 이용하였다. OCL은 일종의 UML확장이다. Semantic Context Broker는 SCE의 문맥인식성, 적응성, 개인화 표현과 기존 PARLAY G/W사이의 브로커(broker)역할을 한다. PARLAY G/W는 서비스 전송 플랫폼을 위한 API(Application Programming Interface) 제공하는 G/W이고, BcN환경에서 신규 망 서비스 제공을 위한 필수적인 요소 기술로 사용된다.

반으로 망 서비스를 개발하고 망 내에서 문맥인식성과 적응성을 제공하는 특징을 가진다.

## II. 개방형 API의 설계

### 2.1 개방형 API를 이용하는 시나리오

개방형 API를 이용하는 시나리오는 다음과 같다. 본 시나리오에서는 Mote나 GPS로부터 무선 디바이스에서의 위치나 속도를 문맥으로 가정하였다. 온라인 게임도중에 게임을 계속하면서 사용자는 밖으로 나가기로 결심한다. 비록 무선 디바이스의 주변에 네트워크 자원이 부족하여 스크린상의 화면 등급의 질이 떨어지더라도 서비스의 중단 없이 계속해서 서비스된다. 즉 망 내부적에서 적응성 있게 대역폭 등을 조정하여 서비스가 중단되지 않도록 한다. 그러므로 사용자는 계속해서 무선 디바이스 상에서 게임을 즐길 수 있다.

### 2.2 기존 PARLAY와 재구성성을 위한 개방형 API의 비교

본 개방형 API 구조는 UML/OCL를 이용하는 SCE와 Semantic Context Broker 그리고 PARLAY G/W로 구성되어 있다. Mote[12]라고 불리는 센서나 GPS(Global Positioning System)등으로부터 문맥을 인식한 후 Semantic Context Broker는 그 문맥을 분석하고 그 문맥에 가장 적합한 서비스를 재구성하여 사용자에게 적응성 있는 서비스를 제공할 수 있다.

정의된 문맥에는 위치(location), 속도(speed), 시간(time), 날씨(weather), 사용자 선호도(User's Preference) 등이 있고, 이러한 문맥은 RCSM, WASP, 개방형 API에서 공통적으로 지원된다. 통신 프로토콜에서 WASP는 웹 상에서 XML을 기반으로 SOAP(Simple Object Access Protocol)을 사용한다. 개방형 API는 PARLAY를 기반으로 SIP(Session Initiation Protocol), SS7(Signaling System 7), 그리고 MAP(Mobile Application Part)등을 지원한다. WASP는 웹 상에서 문맥인식성을 지원한다. 그리고 개방형 API는 PARLAY 기반으로 문맥인식성과 적응성을 지원하는 문맥인식 서비스를 지원한다. 본 개방형 API는 WASP와 유사한 접근방법을 가지고 있지만, 시멘틱 웹 서비스 기술에 독립적이고 통합적인 접근방법인 MDA기

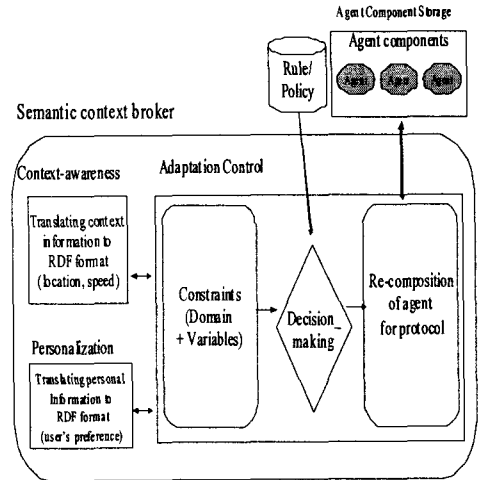


그림 1. 개방형 API를 위한 메커니즘  
Fig.1 Mechanism for the open API

Parlay G/W는 무선망에서는 CAP(CAMEL (Customized Application of Mobile Enhanced Logic) Application Part) 이나 MAP(Mobile Application Part)을 사용하고 IMS(The 3G Multimedia Subsystem)망에서는 SIP(Session Initiation Protocol), PSTN(Public Switched Telephone Network)망에서는 SS7(Signaling System 7) 그리고 유선 망에서는 SIP등을 이용한다. 본 논문에서는 Adaptive Context Broker를 이용하는 PARLAY API를 개방형 API로 명명하였다. (그림 1)은 개방형 API를 위한 메커니즘에 대해서 보이고 있다. 개방형 API는 context-awareness, personalization, Adaptation Control 모듈로 구성되어 있다. Context-awareness 모듈은 위치나 속도와 같은 문맥 정보를 받아서 CC/PP(Composite Capability/ Preference Profile)와 같은 RDF포맷으로 변환하는 역할을 한다. Personalization 모듈은 사용자 선호도(user's preference), 디바이스 타입과 같은 정보를 얻어서 RDF포맷으로 변환하는 역할을 한다. Adaptation control 모듈은 ISP의 정책 등에 따르는 네트워크 프로토콜을 호출하는 에이전트를 재구성한다.

2.3 개방형 API를 위한 문맥(context) 정의

(그림 2)는 개방형 API를 위한 문맥정보, 프로파일, 정책을 보이고 있다. 문맥은 위치, 속도, 날씨, 온도 등을 포함한다. 문맥에 대한 변수 l은 location, s는 speed, wt는 weather, tm은 time of day, p는 presence이다. OCL을 이용한 문맥 제약조건은 다음과 같다. if (current\_location = 'WLAN region') then call RTP protocol else if (current\_location = 'CDMA region') then call WAP protocol 는 현재 위치가 네트워크 자원이 풍부한 WLAN 지역이면 RTP 프로토콜 서비스를 제공하고 현재 위치가 네트워크 자원이 부족한CDMA 지역이면 WAP 프로토콜 서비스를 제공하라는 의미를 가진다.

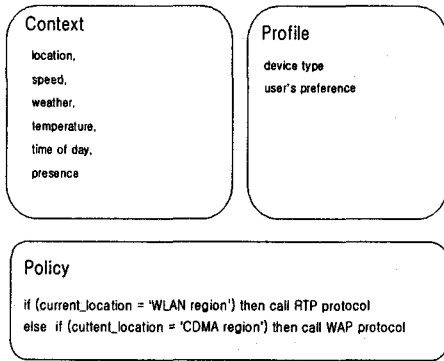


그림 2. 문맥인식을 위한 문맥정보, 프로파일, 정책  
Fig.1 Context Information, Profile, and Policy

III. 개방형 API의 구현

3.1 개방형 API를 위한 스퀀스 다이어그램

(그림 3)은 개방형 API를 위한 시퀀스 다이어그램을 보이고 있다. 센서나 또는 GPS로부터 location과 speed에 대한 문맥을 인식한 후 개방형 API에게 알려준다. 개방형 API는 문맥을 분석 하는데 한 예로 current\_location 이 WLAN 지역인지 CDMA 지역인지를 분석한다. 그 후에 Data Server로부터 사용자는 그 문맥에 적합한 서비스를

지원받는다. 한 예로current\_location이 WLAN 지역이면 RTP 프로토콜 서비스를 지원하고 current\_location이 CDMA 지역이면 WAP 프로토콜 서비스를 제공한다.

구현 환경은 Appium사의 GBOX(13)라는PARLAY X 를 지원하는 소프트웨어 개발 도구를 이용하였다. (그림 7)에서 PARLAY X API를 정의하고 그 API를 위한 간단한 응용을 구현하였다. 본 논문에서는 GBOX라고 불리는 PARLAY X지원 시뮬레이터에서 제공되는 지도 상의 x, y 좌표 값을 설정하여 네트워크 지원이 풍부한 지역과 네트워크 지원이 부족한 지역을 구분하여 설정하였다.

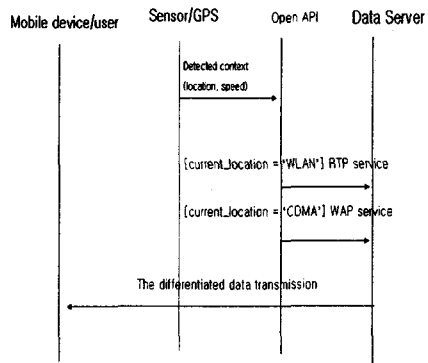


그림 3. 시나리오 시퀀스 다이어그램  
Fig.3 Scenario Sequence Diagram

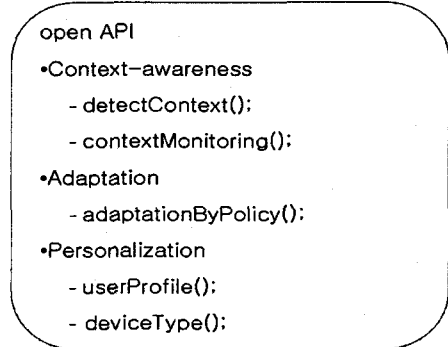


그림4. 정의된 개방형 API  
Fig.4 The defined open API

(그림 4)와 같이 재구성성을 위한 API로는 detectContext(), ContextMonitoring() 이고 적응성을 위해서는 Adapatation ByPolicy()이 있다. 개인화 정보로는 userProfile(), deviceType()이 있다.

## IV. 성능 비교 평가

### 4.1 특성 비교

〈표 1〉에서는 기존 PARLAY X 와 개방형 API를 비교 하였다. 개방형 API는 문맥인식성, 적응성, 개인화를 모두 고려하였다. 그리고 PARLAY 표준을 지원하며UML/OCL 을 포함하는 MDA를 지원한다. 또한 개방형 API에서 Semantic Context Broker은 SCE에서의 OCL을 이용한 문맥인식 언어와 PARLAY G/W인 서비스 전송 플랫폼 사이에 브로커 역할을 수행함으로써, 결과적으로 문맥인식 언어와 PARLAY G/W를 통합 시키는 역할을 한다.

표1. 주된 특성의 비교  
Table 1. The comparison of the main feature

	PARLAY X	개방형 API
QoS (Bandwidth, Latency)	-	X
QoE (MOS)	-	X

### 4.2 성능 비교

PARLAY X, WASP, 그리고 개방형 API 사이에 적응성과 개인화에 대한 특징에 대해서 주로 성능을 비교하였고, 성능비교는 ns-2 시뮬레이터를 이용하였다. ns-2에서 성능 실험을 위해 (그림 3)과 같이 4개의 노드를 설정하였다. 0번 노드는 GPS이고 1번 노드는 데이터 제공 서버이고, 2번 노드는 개방형 API서비스가 실행되는 서버이고 3번 노드는 사용자 단말기로 설정하였다.

성능 실험을 위해서 0번 노드인 GPS는 사용자의 위치를 감지한 후 2번 노드인 개방형 API 실행 서버에게 알려준 경우, 2번 노드인 개방형 API 실행 서버는 사용자의 위치 및 그 위치에 따라서 RTP와 WAP등의 서비스 유형을 결정하고 1번 노드인 데이터 제공 서버는 요청되는 서비스 유형에 따라서 데이터를 전송할 때 사용자에게 전송되는 데이터 패킷의 변화량을 측정하는 실험을 하였다. 서비스 유형 변화 지원을 위해서ns-2를 이용한 tc와 C++코드에서 데이터 전송 도중 서비스 유형이 바뀌는 경우를 위한 ServiceTypeChange()

메소드를 별도로 포함시켰다. ServiceTypeChange()메소드는 개방형 API의 성능 실험을 위해서 정의된 메소드이고, 서비스 유형이 데이터 전송 도중에 변경됨에 따라 패킷 크기가 변경되도록 하는 기능을 지원한다.

(그림 5)에서는 성능 비교를 위해서 개방형 API incoming으로 구성되어 있다. 개방형 API incoming은 개방형 API를 이용한 경우 3번 노드인 사용자 단말기에서 패킷 크기 변화를 측정한 것이고, Existing PARLAY incoming은 기존 문맥인식성 지원 서비스인 경우 3번 노드인 사용자 단말기에서 패킷 크기를 측정한 것이다. 개방형 API를 의미하는 2번 노드인 Semantic Context Broker 실행 서버가 서비스 유형 변경을 지시하면 1번 노드인 데이터 제공서버는 패킷 크기를 RTP service 인 2048에서 WAP service 인 500byte로 조정하면서 서비스를 진행하는 경우, 3번 노드인 사용자 단말기의 패킷 크기 변화를 개방형 API incoming그래프에서 보이고 있다. 반면에서 3번 노드인 사용자 단말기의 Existing PARLAY incoming 그래프는 패킷의 크기가 조정되지 않으므로 중간에 서비스가 중단되는 그래프를 보이고 있다. 그러므로 기존 PARLAY는 적응성에 대한 고려가 부족하기 때문에 망 상황이 변경된 경우 서비스가 중단되었으나 개방형 API는 변화된 환경에 맞게 서비스가 중단되지 않고 지원되므로 본 개방형 API가 적응성을 지원한다는 점을 보인다.

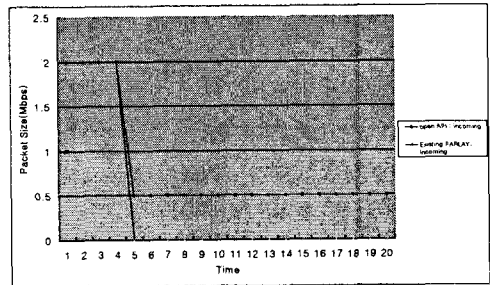


그림 5. 성능 비교  
Fig.5 Performance comparison

## V. 결론 및 향후 연구

본 논문은 망에서 문맥인식성, 적응성, 개인화를 지원하기 위한 PARLAY X기반의 망 서비스 구조와 PARLAY X API를 제안하였고 위치와 속도라는 문맥정보 및 사용자 선호도 정보를 인식하여 망에서 그 문맥정보에 따라서 망에서 각 개인에게 적합한 QoS를 제공하는 경우를 보였다. 본 개방형 API는 기존 PARLAY보다 망 상에서의 문맥인식성, 적응성, 개인화 같은 지능적인 서비스를 지원할 수 있는 장점이 있다. 개방형 API는 향후 보다 다양한 문맥정보에 따라서 망에서 더욱 지능적인 서비스를 제공할 수 있을 것이며, 이러한 망에 지능성을 제공하려는 시도는 제 4세대 무선망이나 차세대 네트워크에 필수적인 기술적인 요소가 될 것으로 보인다. 또한 본 논문의 망 서비스를 위한 구조와 신규 PARLAY API는 PARLAY X표준에 반영될 것으로 기대된다. 향후 연구로는 망에서의 상황인식성(14)에 관한 연구가 필요하다.

## 참고문헌

[1] PARLAY home page, available at [www.parlay.org](http://www.parlay.org).  
 [2] O. Kath, T. Magedanz, R. Wechselberger: "MDA-based Service Creation for OSA/Parlay within 3G beyond Environments," First European Workshop on Model Driven Architecture with Emphasis on Industrial Application, March 17-18, 2004. University of Twente, Enschede, The Netherlands.  
 [3] Aniruddha Gokhale et al. "CosMIC: An MDA Generative Tool for Distributed Real-time and Embedded Component Middleware and Applications," Proceedings of the OOPSLA 2002 Workshop on Generative Techniques in the Content of

Model Driven Architecture, Seattle, WA, November 2002.  
 [4] Bernhard Stenffen, Tiziana Margaria, Andreas Claben, Volker Bruan, "A Constraint-Oriented Service Creation Environment," 2nd International Conference on Practical Application of Constraint Technology, London (UK), April 1996.  
 [5] IKV++ Technologies AG: The medini tool chain, available at <http://www.ikv.de/pdf/mediniWhitePaper.pdf>.  
 [6] C. Efstratiou, K. Cheverst, N. Davies and A. Friday, "An architecture for the effective support of adaptive context-aware applications," Proceedings of 2nd International Conference in Mobile Data Management (MDM'01), Hong Kong, Springer, Vol. Lecture Notes in Computer Science Volume 1987, pp. 15-26, January, 2001.  
 [7] S. S. Yau, F. Karim, Y. Wang, B. Wang, and S. Gupta, "Reconfigurable Context-Sensitive Middleware for Pervasive Computing," IEEE Pervasive Computing, joint special issue with IEEE Personal Communications, 1(3), pp.33-40, 2002.  
 [8] Pareica Dockhorn Costa et. al. "Toward a Service Platform for Mobile Context-Aware Applications," First International Workshop on Ubiquitous Computing (IWUC 2004), Portugal, April 2004.  
 [9] Harry Chen, Tim Finin, Anupam Joshi, "Semantic Web in in the Context Broker Architecture," In Proceedings of PerCom 2004, Orlando FL., March 2004.  
 [10] Norman M. Sadeh, "A Semantic Web Service Environment for Context-Aware Mobile Services," Wireless World Research Forum Conference, Stockholm, September 2001.  
 [11] J. Wing., and Kleppe, A., "OCL: The Constraint Language of the UML," JOOP, May, 1999.  
 [12] L. Girod, J. Elson, A. Cerpa, T. Stathopoulos, N. Ramanathan, D. Estrin, "EmStar: a Software Environment for Developing and Deploying

Wireless Sensor Networks,” in the Proceedings of USENIX General Track 2004.

- [13] Appium Homepage ,SDK supporting PARLAY X, available at <http://www.appium.com/>.
- [14] Weibenberg, "Situation-Aware Service Supply," 2002 Workshop on Generative Techniques in the Content of Model Driven Architecture, Seattle, WA, November 2002.

## 저자 소개

### 홍성준

1967년 8월 22일생  
 1991년 경원대학교 전자계산학과  
 졸업(공학사)  
 1993년 건국대학교 대학원  
 전자계산학과 졸업(공학석사)  
 1998년 건국대학교 대학원  
 컴퓨터공학과 졸업(공학박사)  
 현재 여주대학 정보통신과 조교수  
 <연구분야> 차세대 무선 네트워크,  
 PARLAY X, 시멘틱  
 웹서비스

### 이용수

1963년 10월13일생  
 1986년 명지대학교 전자계산학과  
 졸업(공학사)  
 1989년 건국대학교 대학원  
 전자계산학과 졸업(공학석사)  
 현재 여주대학 컴퓨터정보관리과  
 부교수  
 <연구분야> 데이터베이스, GIS